

REMARKS

In the Office Action mailed January 9, 2008, claims 1-14 and 21-27 were rejected under 35 U.S.C. §103(a) as being obvious over Bengston (U.S. Pat. No. 6, 728,947) in view of Nichols et al. (U.S. Pat. No. 6, 018, 730) in view of Bacon et al. (U.S. Pat. No. 6,430,538) in further view of DeFrancesco Jr. et al. (U.S. Pat. App. Pub. No. 2001/0014877).

The Claims

Claim 22 has hereby been amended to correct a grammatical error. No substantive changes to the claims have been made.

Independent claim 1 recites a workflow management system for hosting process-based tasks and decisioning with a collection of software components on a single platform that includes a software component for business users to establish configurable workflow checklists in real-time in which a plurality of differentiated tasks are set up and are made available for configuring any type of workflow and each workflow task can avail of a plurality of existing or new underlying business parameter objects that can be embedded for workflow task automation, and a data dictionary associated with each workflow. Each workflow is driven by the associated data dictionary for a selected industry to which that workflow corresponds, and the software component for business users has the ability to use, handle and manage the data dictionary and to generate entry conditions and rules dynamically without restarting applications or rewriting underlying software code. The software component for business users includes a graphical interface usable to configure workflows at runtime, which follows a software programming stage, the graphical user interface having a list of business parameter objects represented as geometric shapes and a workspace, each business parameter object represented as a geometric shape being an abstracted object-based representation of functions within the collection of software components, the workspace for organizing and linking multiple geometric shapes at runtime in an ordered arrangement of objects, the ordered arrangement of objects corresponding to an order in which the multiple differentiated tasks are performed when any of the configurable workflow checklists are executed. The system according to independent claim 1 further includes a database for storing the

arrangement of objects in the configurable workflow checklists as well as for storing the entry conditions and embedding information for the business parameter objects that are associated with each of the multiple differentiated tasks.

Independent claim 8 recites a workflow system for programmatically managing dynamic workflow processes that includes a workflow engine, a workflow designer, a rules database, and a data dictionary. The workflow engine is for performing task list processing as defined by a plurality of task lists, with any number of the plurality of task lists processed by the workflow engine at any given time, and is a software component containing a plurality of discrete functions defined for each application within the workflow system prior to runtime. The workflow designer is for configuring a plurality of task lists and has an object-based interface for drag-and-drop creation and modification of task lists at runtime. The workflow designer has a display window that includes a function list containing multiple symbols each corresponding to at least one of the plurality of discrete functions accessible within the workflow engine at runtime, a business parameter object list with each business parameter object therein able to be embedded with any of the discrete functions represented as symbols, a workspace providing a graphical area for assembly of ordered task lists at runtime, and tools for configuring entry conditions associated with any of the plurality of discrete functions for each task list according to logical mathematical operators selected from the rules database and configured at runtime. The workflow designer allows for assembly of ordered tasks by dragging and dropping one of the multiple symbols into the workspace, and embedding business parameter objects with any of the discrete functions represented as symbols, with the workflow designer providing graphical links for assembling and reassembling an ordered task list from multiple discrete symbols. The data dictionary is configurable for each task list for defining discrete data elements and data relationships that are associated with each of the plurality of discrete functions of the workflow engine, and the contents of each data dictionary are specific to a selected industry, and wherein the data dictionaries associated with each task list is dynamically modifiable via the workflow designer in real time without restarting applications or rewriting underlying software programming. As recited by

independent claim 8, the workflow engine performs discrete functions for which all associated entry conditions evaluate to true in an order determined by the ordered task list to render a decision to a remote user.

Independent claim 22 recites a system for programmatically rendering a process-based decision that includes a plurality of configurable discrete tasks made available at runtime, a plurality of business parameter objects made available at runtime and capable of being embedded with any of the plurality of configurable discrete tasks for specifying automation of the process-based decisioning for a checklist, a rules database made available for configuring rule-based entry conditions and selection criteria associated with the configurable discrete tasks at runtime, an administrative interface utilized by business users at runtime for creating process categories and checklists associated with each process and for modifying the entry conditions and the selection criteria associated with the discrete tasks, a decision database for storing the process categories, the checklists, the entry conditions and the selection criteria as configured by business users at runtime, a workflow engine defined on a single platform prior to runtime for automatically processing input from a remote user and generating an instant decision based on the checklist at runtime, a dynamic data dictionary associated with each checklist formatted in XML for defining data elements and data relationships specific to a selected industry, wherein the dynamic data dictionary associated with each checklist provides a dynamic fetch and store interface with the decision database, and wherein the dynamic data dictionary for each checklist is configurable by the business users through the administrative interface at runtime to provide, translate and modify data presentation with respect to both the remote user and the workflow engine such that the workflow engine and the administrative tools can be utilized at runtime by business users across a plurality of industries at runtime without requiring restarting or reprogramming of the administrative interface or the workflow engine to customize the workflow engine and the administrative tools for relevant industries. Independent claim 22 further recites a messaging system for routing two-way communications between the remote user and the process administrator, the messaging system providing a digital record of programmatic transactions.

Independent claim 25 recites a method for workflow processing and programmatic decision-making based on object-based processes stored in memory. The method of independent claim 25 includes defining a plurality of configurable differentiated tasks made available at runtime, defining business parameter objects made available at runtime, defining a rules database containing logical operators for configuring rules-based entry conditions at runtime that are associated with each of the plurality of differentiated tasks, configuring a data dictionary for each of a plurality of process checklists, configuring the plurality of process checklists at runtime, receiving input from a remote source, determining programmatically an input type according to the received input, retrieving automatically a selected one of the plurality of process checklists according to the input type wherein the selected data dictionary acts as an interface between the selected process checklist and the sets of entry conditions and as an interface between the entry conditions and both the data elements and the data relationships as a function of the particular industry to which the received input corresponds, processing programmatically the received information utilizing one or more of the selected set of differentiated tasks based on the entry conditions associated with the stored process checklist, rendering an automatic decision based on the processed received information, communicating programmatically the automatic decision to the remote source and to other partners as specified by the embedded business parameter objects. The data dictionary is populated with data elements specific to a particular industry associated with a selected one of the process checklists and data relationships specific to defined software utilized for processing any of the checklists. According to independent claim 25, the step of configuring each process checklist includes configuring a selected set of the plurality of differentiated tasks in an ordered arrangement, configuring entry conditions associated with each of the selected set of differentiated tasks based upon logical operators from the rules database, and embedding business parameter objects with any of the selected set of differentiated tasks for configuring a degree of automation for the process checklist.

Bengston

Bengston (U.S. Pat. No. 6, 728,947) discloses a workflow distributing apparatus and method. The system of Bengston coordinates a serial, assembly-line style workflow that is executed by a plurality of processing devices. (Bengston, col. 4, line 66 to col. 6, line 3; FIG. 1). The system of Bengston is described with reference to coordinating printing workflows, where steps of the printing process are performed on devices at disparate locations. (E.g., Bengston, col. 1, ll. 10-47; FIG. 3). The workflow is automated so as not to require any user input while in process, and the workflow continues to until completion or until an error occurs. (Bengston, Abstract; col. 11, ll. 5-8 and 27-30; col. 14, ll. 35-39; FIG. 1). Processing by the Bengston system is decentralized, and performed sequentially by a number of different processing devices that push workflows between the processing devices in a predetermined linear fashion. (Bengston, col. 11, ll. 27-31; col. 12, line 58 to col. 13, line 8; FIG. 1). Bengston distinguishes its system from other known systems that utilize centralized processors to control the flow of information. (Bengston, col. 1, line 55 to col. 2, line 4). Bengston does not disclose business parameter objects that are embedded with tasks when configuring workflow checklists for specifying automation of that particular workflow checklist. Also, as noted on page 5 of the Office Action, Bengston fails to show, teach or disclose a data dictionary or entry conditions associated with tasks or functions.

Nichols et al.

Nichols et al. (U.S. Pat. No. 6, 018, 730) discloses a tutorial system installed on and run from a local workstation for helping to teach a student new skills. (Nichols, Abstract; FIG. 1). The system of Nichols et al. provides a simulated environment that students must understand and solve themselves. (Nichols et al., Abstract). Nichols et al. discloses the use of a domain model (or data dictionary) that facilitates communication of context-specific data across generic objects of an application. (E.g., Nichols et al., col. 22, ll. 18-39). The application utilizes a fixed architecture (i.e., workflow checklist) that does not include entry conditions associated with discrete functions in the checklist. Nichols et al., however, does not show, teach or disclose the use of a data dictionaries configured for particular workflow checklists that are dynamically modifiable without

restarting applications or rewriting underlying software programming. Nichols et al also does not disclose business parameter objects that are embedded with tasks when configuring workflow checklists for specifying automation of that particular workflow checklist.

Bacon et al.

Bacon et al. (U.S. Pat. No. 6,430,538) discloses a workflow management system that utilizes personal subflows. Bacon et al. discloses that the personal subflows can specify rule-based branch conditions (or entry conditions), and that work flow activity is determined as a function of whether particular branch conditions evaluate to "true" or "false". (Bacon et al., col. 8, ll. 16-40; col. 9, ll. 27-38; col. 10, ll. 17-45; col. 11, ln. 46 to col. 12, ln. 60). Bacon et al. specifies that the "personal subflow does not have any explicit definition of participants or agents. Instead this information is linked or bound at run-time for the personal subflow, not at definition time." (Bacon et al., col. 3, ll. 33-36; col. 9, ll. 1-6; col. 13, ll. 1-3). In other words, workflow tasks or functions (i.e., the personal subflows) are not defined prior to runtime. Instead, Bacon et al. discloses that the personal subflows are limited constructs that are unable to associate processing activity with work items, or perform activities using more than one participant. (Bacon et al., col. 9, ll. 14-19). Moreover, Bacon et al. lacks embedded business parameter objects.

DeFrancesco Jr. et al.

DeFrancesco Jr. et al. (U.S. Pat. App. Pub. No. 2001/0014877) discloses a workflow management system for an automated credit application system. The workflow management system of DeFrancesco Jr. et al. includes rule elements 306 associated with one or more boolean tests 308, which are used "to determine if a particular processing step is complete, or can be skipped." (DeFrancesco Jr. et al., ¶¶ 44, 45, 52, 53; FIGS. 2 and 3; see also ¶¶ 69-113 and the claims). Workflow elements are associated in a workflow definition 404 and stored in a relational database management system (DBMS) 104. (DeFrancesco Jr. et al., ¶¶ 48, 59, 60; FIG. 4). The status of workflow steps is tracked with flags generated as a function of the tests 308. (DeFrancesco Jr. et al., ¶67). A "workflow configuration tool [402] is used at run-time to define

customized workflow requirements[, which] . . . alleviates that need to customize source code for each client and each time workflow requirements change." (DeFrancesco Jr. et al., ¶¶ 22, 58).

Claim Rejections – 35 U.S.C. §103(a)

Claims 1-14 and 21-27 were rejected under 35 U.S.C. §103(a) as being obvious over Bengston (U.S. Pat. No. 6, 728,947) in view of Nichols et al. (U.S. Pat. No. 6, 018, 730) in view of Bacon et al. (U.S. Pat. No. 6,430,538) in further view of DeFrancesco Jr. et al. (U.S. Pat. App. Pub. No. 2001/0014877). However, Bengston in view of Nichols et al. in view of Bacon et al. in further view of DeFrancesco Jr. et al. fail to disclose, teach or suggest each and every limitation of the pending claims. For instance, each workflow task can avail of a plurality of existing or new underlying business parameter objects that can be embedded for workflow task automation as recited in claim 1, business parameter objects are embedded with any of the discrete functions represented as symbols as recited by claim 8, a plurality of business parameter objects are made available at runtime and are capable of being embedded with any of the plurality of configurable discrete tasks for specifying automation of the process-based decisioning for a checklist as recited by claim 22, and business parameter objects are defined and made available at runtime and are embedded with any of the selected set of differentiated tasks for configuring a degree of automation for the process checklist as recited by claim 25. In each of independent claims 1, 8 22 and 25, the business parameter objects are distinguished from the workflow tasks (claim 1)/plurality of configurable discrete tasks (claim 8)/selected set of differentiated tasks (claim 25). The Office Action cites passages and figures of Bengston that disclose icons 30 representing process steps. However, these process steps are comparable only to the present workflow tasks (claim 1)/plurality of configurable discrete tasks (claim 8)/selected set of differentiated tasks (claim 25), but fail to disclose, teach or suggest business parameter objects that are embedded within process steps/tasks/functions. As recited in claims 1, 22 and 25, the embedded business parameter objects specify automation for a workflow or checklist. Indeed, claim 1 further recites that each business parameter object represented as a geometric shape is an abstracted object-based representation of functions within the collection of software components.

The Office Action cites col. 10 of Bengston at disclosing “business parameters”, but there Bengston merely discloses data fields (e.g., numerical values) linked to specific process steps. Mere numerical entries do not satisfy the “business parameter object” limitations of the present claims. The other cited references do not supply these missing limitations.

Furthermore, claim 25 recites communicating programmatically the automatic decision to the remote source and to other partners as specified by the embedded business parameter objects. The Office Action applied the same rejections of claims 1-8 because they recite the same limitations. However, this limitation of claim 25 is not found in claims 1-8. The cited references fail to disclose that embedded business parameter objects specify programmatic communication to multiple parties.

The present claim rejections also rely on hindsight reasoning, and lack a motivation to combine the cited references in the manner suggested in the Office Action. In order to establish a *prima facie* case of obviousness under 35 U.S.C. §103, the rejections must rest on a factual basis. In making such a rejection, the examiner has the initial duty of supplying the requisite factual basis, and may not rely upon speculation, assumption or hindsight reconstruction to supply deficiencies in the factual basis. (*In re Warner*, 37 F.2d 1011, 1017 (CCPA 1967), *cert denied*, 389 U.S. 1057 (1968)). Furthermore, obviousness rejections "cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness." (*KSR Int'l Co. v. Teleflex, Inc.*, 550 U.S. ___, 82 USPQ.2d 1385 (2007) (*quoting In re Kahn*, 441 F. 3d 977, 988 (Fed. Cir. 2006))). This reasoning should be made explicit, and it is "important to identify a reason that would have prompted a person of ordinary skill in the relevant field to combine the [cited prior art] elements in the way the claimed new invention does. This is so because inventions in most, if not all, instances rely upon building blocks long since uncovered, and claimed discoveries almost of necessity will be combinations of what, in some sense, is already known." (*KSR Int'l Co. v. Teleflex, Inc.*, 550 U.S. ___ (2007); *see also* M.P.E.P. §2142). The M.P.E.P. warns against the improper use of hindsight in attempting to establish *prima facie* obviousness.

To reach a proper determination under 35 U.S.C. 103, the examiner must step backward in time and into the shoes worn by the hypothetical “person of ordinary skill in the art” when the invention was unknown and just before it was made. In view of all factual information, the examiner must then make a determination whether the claimed invention “as a whole” would have been obvious at the time to that person. Knowledge of applicant’s disclosure must be put aside in reaching this determination, yet kept in mind in order to determine the “differences,” conduct the search and evaluate the “subject matter as a whole” of the invention. The tendency to resort to “hindsight” based upon applicant’s disclosure is often difficult to avoid due to the very nature of the examination process. However, impermissible hindsight must be avoided and the legal conclusion must be reached on the basis of the facts gleaned from the prior art.

MPEP §2142.

Further, if the proposed modification of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the cited references are not sufficient to render a claim *prima facie* obvious. *In re Ratti*, 270 F.2d 810 (CCPA 1959).

Regarding the motivation to combine the four cited references, the Office Action at page 6 states that together “they streamline the use of automated tools reducing overall costs[,]” “which is a goal of Bengston” However, this assertion does not answer the question of why someone of ordinary skill in the art would have known to pick and choose certain elements of each of the four cited references, adopting some features from each reference but disregarding others. Moreover, the modifications necessary to combine the cited references in the manner suggested by the Office Action would change the principle of operation of many of those references. For instance, while Bengston does discuss “reducing cost [and] . . . automatic otherwise manual operations . . . ,” Bengston discloses a system that operates on a fundamentally different principles of operation than some of the other references cited in combination by the Office Action. The workflows of Bengston and Nichols et al. are predefined workflows that lack dynamic processing capabilities provided via entry conditions associated with tasks available in the workflow and the

ability to modify workflows during runtime without restarting or re-programming underlying software. The Office Action suggests modifying the Bengston and Nichols et al. references according to the teachings of DeFrancesco Jr. et al., but there is a fundamental difference in operational principle between a system that utilizes a fixed, entirely pre-coded workflow processing scheme and one that provides for dynamic configuration at runtime without modification to underlying code.

A rather vague and general desire for efficiency and cost-effectiveness is not sufficient to establish a *prima facie* case of obviousness where the proposed modification involves changing a basic principle of operation that a system follows. To make an analogy, the fact that lumber and screws are well known does not render every possible structure built from lumber and screws obvious, and likewise, it does not automatically follow from the fact that certain systems include software-implemented elements are individually known that any and all possible systems that utilize some of those components are obvious. As noted by the Supreme Court in the *KSR* case discussed above, there must be “a reason that would have prompted a person of ordinary skill in the relevant field to combine the [cited prior art] elements *in the way the claimed new invention does.*” (*KSR Int'l Co. v. Teleflex, Inc.*, 550 U.S. ____ (2007) (emphasis added). The mere existence in the prior art of component parts of a claimed system is insufficient to establish a *prima facie* case of obviousness. Rather, there must be a reasoned explanation somewhere in the prior art or the knowledge generally available to one of ordinary skill that establishes how and why various elements would be combined *in the particular way claimed.*

The Office Action relies upon hindsight reasoning to impermissibly combine the teachings of Bengston, Nichols et al., Bacon et al. and DeFrancesco Jr. et al. in a way that is only suggested by the disclosure of the present invention. Moreover, the modifications suggested in the Office Action in order to combine the cited art to arrive at the present invention as claimed would impermissibly require a change in the basic principles of operation of certain references.

Thus, the rejections of independent claims 1, 8, 22 and 25 under §103 should be withdrawn. Notification to that effect is requested.

Claims 2-7 and 21 depend from claim 1 and include all of the limitations of that base claim, claims 9-14 depend from claim 8 and include all of the limitations of that base claim, claims 23 and 24 depend from claim 22 and include all of the limitations of that base claim, and claims 26 and 27 depend from claim 25 and include all of the limitations of that base claim. Therefore, for the reasons given above with respect to independent claims 1, 8, 22 and 25, dependent claims 2-7, 9-14, 21, 23, 24, 26 and 27 are likewise allowable over the cited art, and the rejections of those claims should be withdrawn.

CONCLUSION

All of the pending claims are in condition for allowance. The Commissioner is authorized to charge any additional fees associated with this paper or credit any overpayment to Deposit Account No. 11-0982.

Respectfully submitted,

KINNEY & LANGE, P.A.

Date: 2.26.2008

By:



Austen Zuege, Reg. No. 57,907
THE KINNEY & LANGE BUILDING
312 South Third Street
Minneapolis, MN 55415-1002
Telephone: (612) 339-1863
Fax: (612) 339-6580

AZ:kmm